CHAPTER 8

True or False?

What is the difference between the following two questions:

- What color is the sky?
- Is the sky blue?

Well, there are a *lot* of differences: the first question has five words and the second one has four words, for example. My real point in asking these two questions is to point out that the first question is open-ended; the sky could be blue, or gray, or any number of answers. The second question, however, only has two possible answers: yes or no.

When it comes to programming your robots, you need to understand that many times your robots can only *provide* you with those two answers: yes or no. At other times, your robots can *understand* only a yes or no answer. Understanding how to take a yes/no answer and use it to properly program your bots is the focus of this short chapter.

One or the Other

Let's have a question/answer session with SPOT:

Me: SPOT, what color is the box in front of you?

[SPOT sits there and gives no response.]

Me: SPOT, what is the position of your Touch sensor button?

[SPOT still sits there and gives no response.]

Hmmm . . . SPOT doesn't seem to be to responsive today. I seem to remember, however, that SPOT prefers yes/no questions, so let me try this again:

Me: SPOT, is the color of the box in front of you blue?

SPOT: Yes *[appears on the LCD screen]*

Me: SPOT, is your Touch sensor button pressed?

SPOT: No *[appears on the LCD screen]*

OK, now we're getting somewhere. SPOT does prefer to communicate with me using yes or no answers. Another way of saying this is that SPOT prefers to communicate using *logical responses*; a logical response is simply Yes or No.

■ **Note** Some computers and robots use True or False, but it's all the same: Yes = True and No = False. Some computers and robots even use 1 or 0 as a logical response, where 1= True and 0 = False. There's even another method: On or Off! In that case, On = True and Off = False. But for the purposes of this chapter and programming, let's stick with either Yes/No or True/False.

Let's have another conversation with SPOT:

Me: SPOT, is your Ultrasonic sensor detecting an object 6 inches in front of you?

SPOT: True.

Me: SPOT, is your Right button being pressed?

SPOT: False.

Apparently, SPOT's sensors have the ability to return a logical response to SPOT that he can pass along to me. SPOT listens to his sensors' conditions and responds with True or False.

What does all this have to do with programming, though? Here's your answer: your NXT robots can send and receive logical responses to and from the sensors, motors, buttons, and other items.

As an example, take a look at Figure 8-1.

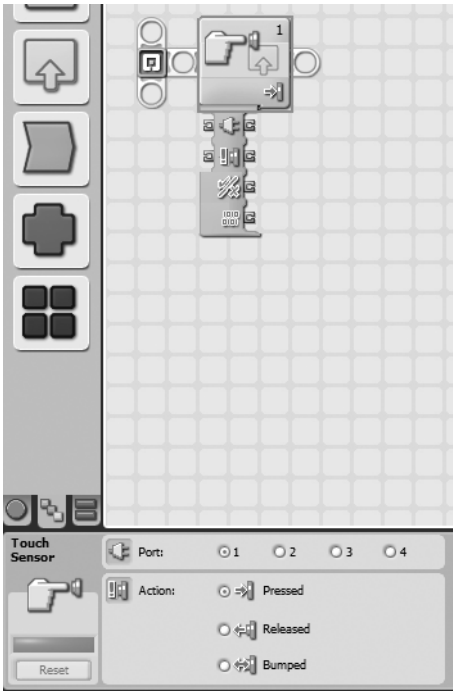

Figure 8-1. *The Touch sensor's configuration panel*

What you are looking at in Figure 8-1 is the TOUCH SENSOR block and its configuration panel. Notice also that the TOUCH SENSOR block has its data hub visible (see Chapter 7 for a discussion on data hubs and Chapter 9 for a more detailed discussion of the sensors). If you are only seeing one data output plug on the data hub, click the data hub again to expand it to its full size.

On the data hub, you'll see some small input and output data plugs. If you hover the mouse pointer over the third plug from the top, the words "Yes/No" will appear. What this tells you is that this data plug can provide output (using a data wire) in the form of a Logic data type.

But how do you know if the output will be Yes or No? Simple—the answer is based on what you are monitoring with the sensor.

In Figure 8-1, notice that the Touch sensor's Action section has the Pressed option selected. This means that if the Touch sensor's button is pressed down (and not released) the Yes/No Logic plug would provide a Yes answer. If the button is not pressed, the Yes/No Logic plug will provide a No answer.

Think back to Chapter 7; I told you that when you connect two blocks with a wire, the input and output data plugs *must* be carrying the same data type. In this case, if you wish to connect a wire from the Yes/No data plug, it must be going into a block that has a Logic data type input plug.

As an example, take a look at the new block in Figure 8-2.

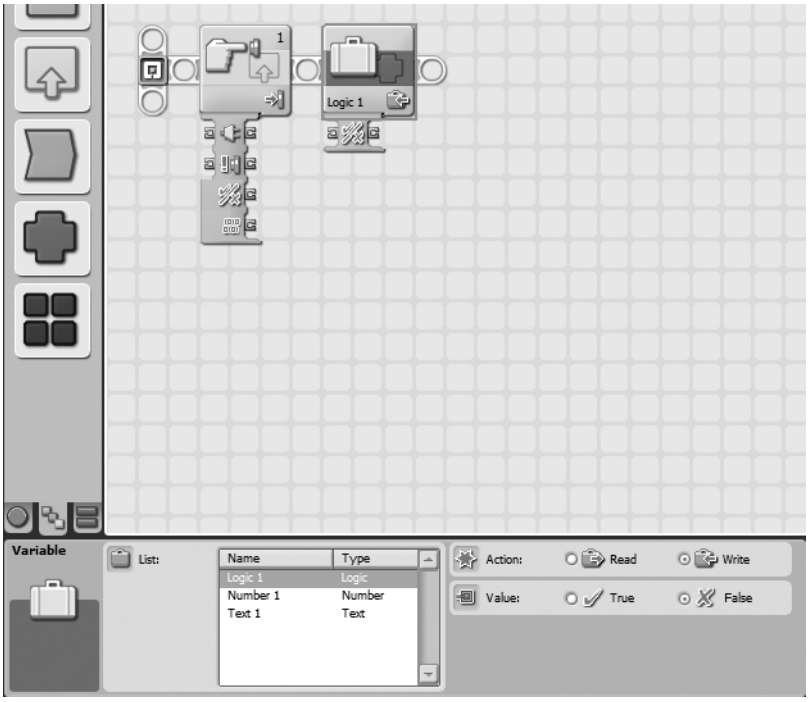

Figure 8-2. *I've dropped a VARIABLE block on the beam.*

This new block is a VARIABLE block. I cover this block in more detail in Chapter 18, but for now all you need to know about the VARIABLE block is that it can hold one of the three data types: Logic, Number, or Text. In Figure 8-2, I've configured the VARIABLE block to hold a Logic

value. I've also opened the data hub, so you can see that it has a Yes/No input data plug. All that I need to do is connect it to the TOUCH SENSOR block with a wire (see Figure 8-3).

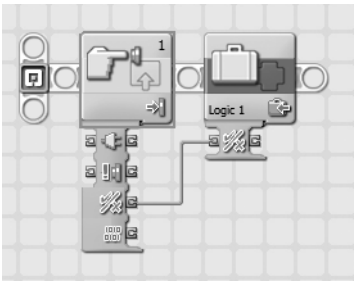

Figure 8-3. *Connecting two blocks with a wire*

When connecting blocks in a program using data wires, always keep in mind that a data wire will *only* work if it is connected to an input and output plug that expect the same data type (Logic, Number, or Text). I also need to point out that many blocks hold either a True or False value as a default setting. For example, the VARIABLE block in Figure 8-2 is configured to hold a default value of False. But you could easily change this to True (I've got more examples later in this book where you'll configure True/False values).

The Logic data type can be found in many blocks, especially the sensor blocks (see Figure 8-4). The sensor blocks all have a data plug that provides a Yes/No response. These plugs are designated by a check mark and an "X" that symbolize the Yes/No response.

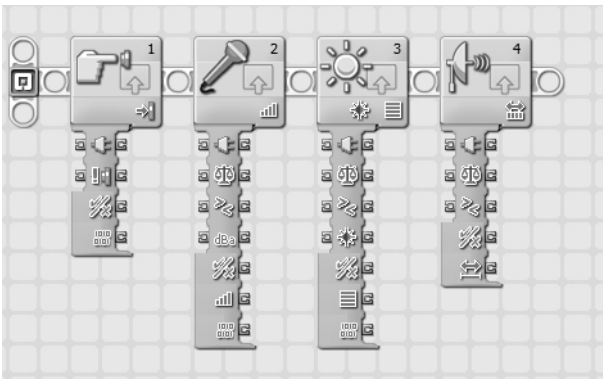

Figure 8-4. *The four sensors (Touch, Sound, Light, and Ultrasonic) all have Logic data plugs.*

Where the Logic data type really comes in handy, however, is with the LOOP and SWITCH blocks (these are covered in Chapters 11 and 12, respectively). Logic data types are very useful when programming a bot to make decisions on its own, and the LOOP and SWITCH blocks can both use a Yes/No response (as input) to give your robots more complex behaviors and control of itself. The bot can examine a sensor or motor or other type of input and, based on the Yes/No response, make further decisions about what it does next.

And that's it for this short chapter on logic. As I said, you'll get a more detailed description of how to use the Logic data type with the LOOP and SWITCH blocks in Chapters 11 and 12. Now let's change direction in Chapter 9 and take a look at some more items that can be used to communicate with the NXT Brick: sensors, buttons, and timers.